

In the Specification

Please delete paragraph [0038] in its entirety and replace with the following:

[0038] In the present embodiment, a second rules engine 28 has the role of acting on some or all of the data cache or other management rules. To support this role, the second rules engine 28, according to the present embodiment, has an IMDBMS plug-in 14d and a custom data retriever plug-in 18d. The second rules engine 28 has access to the rules definitions, which may include that complete instruction set for acting on the data, stored in cache, for example, during initialization of each of the application servers 10a through 10c via its IMDBMS plug-in 14a-14c. The IMDBMS 20 keeps all in-memory cache consistent or synchronized. Data that is present in the cache is visible to application server 10a is also visible to the cache and visible to the second rules engine 28, as well as to the other application servers 10b through 10c. It should be appreciated that the actual memory location where the data is cached may be local to the application server 10a, the IMDBMS 20 or on other systems provided with such memory or storage capability.

Please delete paragraph [0039] in its entirety and replace with the following:

[0039] According to one aspect, the IMDBMS 20 contains a rules event table 25 whose entries include a rule type or rule event and a reference to the data associated with the rule type, such as a pointer. The rule type identifies a rule definition which fully describes the function to be applied to the data. The rule event or type entry is provided by the wrapper 12 when a data item is first entered into the cache by an application. In this embodiment, the second rules engine 28 is operable to periodically poll or query, based on the rules, the IMDBMS 20. In this manner, the second rules engine 28 identifies cached data with the associated rule type. After

identifying the cached data and the rule type, the second ~~[[rule]]~~ rules engine 28 applies the rule to the related data, thereby implementing the application 11a specific rule outside of the IMDBMS 20. It can be seen that this functionality provides implementation of application 11a specific rules without the inefficiencies associated with these rules being implemented by the application 11a. In addition, the present embodiment provides cache management of data functionality not provided by the IMDBMS 20, which promotes greater efficiency throughout the system.

Please delete paragraph [0040] in its entirety and replace with the following:

[0040] The second ~~[[rule]]~~ rules engine 28 may monitor cache data by polling the rules event table 25 and data items in the cache. When a rule associated with a data item is ready to be executed, the second ~~[[rule]]~~ rules engine 28 looks up the associated rule definition, accesses the data item, and performs the cache management operation defined by the rule upon the data item. The wrapper 12 may build an event table in the IMDBMS 20 which links data items with rule types. The rule type may be associated with rule definitions. The second ~~[[rule]]~~ rules engine 28 may monitor this rule event table by polling the table. The second ~~[[rule]]~~ rules engine 28 may be distinguished from the rules engine 17 by this polling activity. In other aspects, the rules engine 17 and second ~~[[rule]]~~ rules engine 28 may be combined.

Please delete paragraph [0041] in its entirety and replace with the following:

[0041] In one embodiment the second ~~[[rule]]~~ rules engine 28 may run on a dedicated general purpose computing system, which will be discussed in detail hereinafter. In another embodiment the second ~~[[rule]]~~ rules engine 28 may share a general purpose computing system with an application server or other programs. In another embodiment the second ~~[[rule]]~~ rules engine 28 may share a general purpose computing system with the IMDBMS 20.

Please delete paragraph [0046] in its entirety and replace with the following:

[0046] The IMDBMS plug-in 14a provides the wrapped data 49-1 to the IMDBMS 20, which stores the data in cache and associates a component of the rule, such as a rule type, to the data and stores the rule type in the [[rule]] rules event table 25 or other location. The second [[rule]] rules engine 28 periodically queries 49-2 the IMBDMS server 20 to apply the rule to the cached data in the IMDBMS 20. Note that these two messages – 49-1 and 49-2 – are not part of the synchronous data management action, but are shown to aid comprehension of system operation. Message 49-1 occurs as part of the normal operation of the IMDBMS 20 maintaining synchronization of data cache across all cache users. Message 49-2 is an on-going message the second [[rule]] rules engine 28 periodically sends to the IMDBMS 20.

Please delete paragraph [0047] in its entirety and replace with the following:

[0047] If the application 11a requests that a data item be refreshed from the back office DBMS 22 (data rule 2), the wrapper 12 requests that the rules engine 17 read the data item out of the back office DBMS 22 via the services of the custom data retriever plug-in 18a. The rules engine 17 requests the data from the custom data retriever plug-in 18a. The custom data retriever plug-in 18a reads the data out of the back office DBMS 22 and returns the data to the rules engine 17. The rules engine 17 returns the data to the wrapper 12. The wrapper 12 writes the returned data – updated with appropriate rule index, date and time, and supplemental information – into cache and returns the unwrapped data item to the application 11a. In another embodiment the wrapper 12 may signal the second [[rule]] rules engine 28 so that the second [[rule]] rules engine 28 may read the data item out of the back office DBMS 22 via the services of its custom data retriever 18d. The second [[rule]] rules engine 28 then writes the returned data – updated with appropriate date and time and

optional supplemental information – into cache. The wrapper 12 then reads the data item out of cache, strips off the rule and optional supplemental information, and returns the simple data to the application 11a. This is said to be a synchronous data manipulation because the data manipulation is performed in direct response to an application action.

Please delete paragraph [0048] in its entirety and replace with the following:

[0048] If the application 11a marks a data item for removal from cache (data rule 7), the wrapper requests the rules engine 17 to remove the data from cache. The rules engine 17 then removes the data from cache and removes the rules table entry associated with the data item from the rules table. In another embodiment, the wrapper 12 may signal the second rules engine 28 so that the second rules engine 28 deletes the data item from cache and remove the rules event table 25 entry associated with the data item from the rules event table 25. This is said to be a synchronous data manipulation because the data manipulation is performed in direct response to an application action.

Please delete paragraph [0050] in its entirety and replace with the following:

[0050] The remaining data manipulation rules may be described as asynchronous, and the associated data manipulations may be handled, according to the present embodiment, by the second rules engine 28. The second rules engine 28 has visibility to cached data via its own IMDBMS plug-in 14d and monitors this cache for changes. When a cached data item is modified, the second rules engine 28 has access to this information. This access may be through a database trigger mechanism if the IMDBMS 20 supports trigger mechanisms, or the second rules engine 28 may periodically poll data items.

Please delete paragraph [0051] in its entirety and replace with the following:

[0051] When a tenure of a data item expires (rules 3-6) the second ~~[[rule]]~~ rules engine 28 removes the data item from cache and removes the rules table entry associated with the data item from the rules table. If rules 5 or 6 are invoked, the second ~~[[rule]]~~ rules engine 28 writes the data back to the back office DBMS 22 via its custom data retriever plug-in 18d. If a data item is governed by rule 1 and is due for its periodic refresh, the second ~~[[rule]]~~ rules engine 28 reads the data item from the back office DBMS 22 via its custom data retriever plug-in 18d and then writes the data item into its cache.

Please delete paragraph [0053] in its entirety and replace with the following:

[0053] The software applications described above may be implemented on any general-purpose computer with sufficient processing power, memory resources, and network throughput capability to handle the workload placed upon it. Figure 3 illustrates a typical, general-purpose computer system suitable for implementing the present invention. The computer system 50 includes a processor 52 (which may be referred to as a central processor unit or CPU) that is in communication with memory devices including secondary storage 54, read only memory (ROM) 56, random access memory (RAM) 58, input/output (I/O) ~~[[60]]~~ devices 60, and network connectivity devices 62. The processor may be implemented as one or more CPU chips.

Please delete paragraph [0055] in its entirety and replace with the following:

[0055] I/O ~~[[60]]~~ devices 60 may include printers, video monitors, keyboards, mice, track balls, voice recognizers, card readers, paper tape readers, or other well-known input devices. The network connectivity devices 62 may take the form of modems, modem banks, ethernet cards, token ring cards, fiber distributed data interface (FDDI) cards, and other well-known network devices. These

network connectivity 62 devices may enable the processor 52 to communicate with an Internet or one or more intranets. With such a network connection, it is contemplated that the processor 52 might receive information from the network, or might output information to the network in the course of performing the above-described method steps. Such information, which is often represented as a sequence of instructions to be executed using processor 52, may be received from and outputted to the network, for example, in the form of a computer data signal embodied in a carrier wave.

Please delete paragraph [0057] in its entirety and replace with the following:

[0057] Turning now to Figure 4 the location on computers of programs cooperating in one embodiment is depicted. The application server 10a is executing on general computer system #1 80. The back office DBMS 22 executes on general computer system #2 82. The second ~~[[rule]]~~ rules engine 28, which, in some embodiments, may include functionality of rules engine 17, is executing on general computer system #3 84. The IMDBMS 20 is executing on general computer system #4 86. The application server 10c is executing on general computer system #5 88. Note that some of these general computer systems may be several computers networked together to increase the collective processing power.

Please delete paragraph [0059] in its entirety and replace with the following:

[0059] Turning now to Figure 6 the location on computers of programs cooperating in another embodiment is depicted. The application server 10a and the second ~~[[rule]]~~ rules engine 28 are executing on general computer system #1 80. The back office DBMS 22 executes on general computer system #2 82. The IMDBMS 20 ~~are both~~ is executing on general computer system #3 84. The application server 10c is executing on general computer system #4 86. Note that some of these

general computer systems may be several computers networked together to increase the collective processing power.

Please delete paragraph [0060] in its entirety and replace with the following:

[0060] Turning now to Figure 7 the location on computers of programs cooperating in another embodiment is depicted. The application server 10a, IMDBMS 20, and the second [[rule]] rules engine 28 are executing on general computer system #1 80. The back office DBMS 22 executes on general computer system #2 82. It should be understood that it is within the spirit and scope of the present disclosure that the rules engine 17 and second [[rule]] rules engine 28 are combined, separated or distributed as necessary to achieve desired functionality and efficiency. The application server 10c is executing on general computer system #3 84. Note that some of these general computer systems may be several computers networked together to increase the collective processing power.

In the Drawings

Applicant submits concurrently herewith, two (2) ***Replacement Sheets***, Figures 1 and 7. The enclosed ***Replacement Sheets*** supersede the original drawings filed by Applicant on July 28, 2003, replacement drawings filed on July 30, 2007.